



TITLE:

位置情報をもつグラフデータに対する知識表現言語: Layout Formal Graph System (計算機科学の基礎理論: 21世紀の計算パラダイムを目指して)

AUTHOR(S):

糸川, 裕子; 内田, 智之; 宮原, 哲浩; 中村, 泰明

---

CITATION:

糸川, 裕子 ...[et al]. 位置情報をもつグラフデータに対する知識表現言語: Layout Formal Graph System (計算機科学の基礎理論: 21世紀の計算パラダイムを目指して). 数理解析研究所講究録 2000, 1148: 158-163

ISSUE DATE:

2000-04

URL:

<http://hdl.handle.net/2433/63997>

RIGHT:

# 位置情報をもつグラフデータに対する知識表現言語 Layout Formal Graph System

糸川裕子 (Yuko Itokawa) 内田智之 (Tomoyuki Uchida)  
宮原哲浩 (Tetsuhiro Miyahara) 中村泰明 (Yasuaki Nakamura)

広島市立大学大学院 情報科学研究科  
{yuko@toc.cs, uchida@cs, miyahara@its, nakamura@cs}.hiroshima-cu.ac.jp

## 概要

位置情報をもつグラフ構造データから知識を発見するシステムの知識表現言語として、新しいグラフ生成システム Layout Formal Graph System(LFGS)を提案する。LFGSは、Formal Graph System(FGS)に位置情報を加味したグラフ文法である。また、発見システムにおける知識表現言語としてのLFGSの有効性を示すために、LFGSで項として用いるグラフの同型問題を解く多項式時間アルゴリズムを与える。さらに、LFGSと既存のグラフ文法であるレイアウトグラフ文法(LGG)とを比較し、LFGSの方がLGGよりも生成能力が高いことを示す。

## 1 はじめに

いくつかの点とそれらを結ぶ線からなる図形によって表すと便利である状況がしばしば見られる。このような関係を数学的に抽象化した概念をグラフという。分子構造やCAD、ネットワークなどは、頂点が概念やオブジェクトを表し、辺はそれらの関係を表すグラフで抽象化することができる。一般に、グラフは事象(頂点)とその関係を表すだけで、事象間の距離は考慮されない。しかし、事象の位置が重要な画像データなどをグラフで表すとき、そのグラフを距離空間上で定義する必要がある。

本研究の目的は、距離空間上のグラフから効率良く知識を獲得するシステムの設計である。従来から、機械発見や発見科学といわれるような分野では、いろいろなデータを分析し、計算機によって何らかの新しい知識を発見する研究が行われてきた。現在でも文字列を扱う発見システムは多く存在する[1]が、2次元や3次元空間上でグラフ構造をしているような事象の位置関係が重要なデータを扱う発見システムはあまり多く存在しない。事象の位置が極めて重要な意味をもつ、地図データやゲノムデータから知識を得ようとする場合、その知識を表現するためのグラフおよび規則を距離空間上で定式化しなければならない。

本論文では、この知識獲得システムでの出力である仮説(知識)を表現するために、Layout Formal Graph System(LFGS)を定義する。これは、グラフを項としてもつ論理プログラミングシステムであるFormal Graph System(FGS)[6]に位置情報を加味した規則であり、距離空間上のグラフを扱うことができる。

さらに、LFGSで用いるレイアウト項グラフでの同型問題について考察する。一般のグラフ同型問題はNPに入っているがPに入っているかは未解決な問題の一つである。しかし、本論文で定義したレイアウト項グラフにおける同型問題は、閉路をもたない有向グラフ(directed acyclic graph, DAG)のハミルトンパスを見つける問題に帰着することができる。閉路をもたない有向グラフのハミルトンパスを見つける問題は多項式時間で解けることが知られている[4]。この多項式時間アルゴリズムを用いることによって、レイアウト項グラフの同型問題は多項式時間で解けることを示す。

最後に LFSG とレイアウトグラフ文法 (Layout Graph Grammar, LGG)[2] とを比較する. LGG  $G$  が与えられたとき,  $G$  によって生成されるグラフ  $g$  の導出過程を模倣する LFSG  $\Gamma_G$  を構築し, そのグラフ  $g$  が  $\Gamma_G$  により生成可能であることを示す. これにより, LGG で生成できるグラフの族 (木, 完全グラフ, 二部グラフなど) は LFSG でも生成することができる. さらに, 帰納論理プログラミング (Inductive Logic Programming, ILP)[3] という学習技法を用いた発見システムが提案されている [5] が, 論理プログラミングで定義された LFSG を知識表現としてもち, ILP を学習手法としてもつ発見システムを構築することができる.

## 2 Layout Formal Graph System

文字列を対象とした文脈自由文法概念をグラフに拡張して得られた生成文法として, 頂点ラベル制御グラフ文法や超辺置き換えグラフ文法などが知られている. 本研究では文脈自由グラフ文法よりもグラフ生成能力が高い Formal Graph System (FGS)[6] を扱う. FGS は距離空間上に表されたグラフを対象としていない. そこで, Brandenburg によって提案された描画仕様 [2] をもとに FGS を拡張し, 距離空間上のグラフを対象とする Layout Formal Graph System (LFGS) を新たに構築する.

$\Sigma, \Lambda$  を有限アルファベット,  $X$  を変数の集合とする.  $\Sigma, \Lambda \cup \{x, y\}$ ,  $X$  の各要素をそれぞれ頂点ラベル, 辺ラベル, 変数ラベルという.

$V$  頂点集合. 頂点のラベル付け写像を  $\varphi: V \rightarrow \Sigma$  とする.

$E$  有向辺集合  $E \subseteq V \times (\Lambda \cup \{x, y\}) \times V$ . 辺  $(u, a, v) \in E$  に対し,  $\psi((u, a, v)) = a \in \Lambda \cup \{x, y\}$  であるラベル付け写像を  $\psi: E \rightarrow \Lambda \cup \{x, y\}$  とする.

$H$  要素が超辺と呼ばれる,  $V$  の 1 つ以上の頂点のリストの族 (要素の重複を許す集合). 超辺は写像  $\lambda: H \rightarrow X$  で辺ラベルがつけられている. 超辺  $h \in H$  の要素をポートといい,  $h$  のポート全体の集合を  $V(h)$  と表す.

$F$  超辺と頂点または超辺間の辺の集合  $F \subseteq \{(h, s, v), (v, s, h) \mid v \notin V(h), v \in V, h \in H, s \in \{x, y\}\} \cup \{(h, s, h') \mid h, h' \in H, h \neq h', s \in \{x, y\}\}$ . 各辺は写像  $\psi': F \rightarrow \{x, y\}$  でラベルづけされている.

$dist$   $\psi(e) \in \{x, y\}$  である辺  $e \in E \cup F$  に非負整数を割り当てる関数.

$perm$   $h \in H$  に対し  $h$  の要素を並べ替えたリストへの写像  $perm: H \rightarrow V^*$ .

$E, F$  中のラベル  $s \in \Lambda \cup \{x, y\}$  をもつすべての辺の集合をそれぞれ  $E^s, F^s$  と表す. さらに,  $E^s, F^s$  により誘導される  $g$  の部分グラフをそれぞれ  $g[E^s], g[F^s]$ , それらの頂点集合をそれぞれ  $V(E^s), V(F^s)$  と表す. 変数に非負整数を割り当てる関数を  $rank: X \rightarrow N$  とする. これにより, 変数  $x$  のポートの数を与える. また, 集合またはリスト  $A$  の要素数を  $|A|$  で表す. このとき, 次の条件 1-3 を満たす組  $g = (V, H, E, F, dist, perm)$  をレイアウト項グラフ (layout term graph) という:

- (1)  $V(E^x) = V(E^y) = V$  であり, かつ  $g[E^x]$  および  $g[E^y]$  において任意の 2 頂点間には必ずパスが存在する. また,  $g[E^x \cup F^x]$  および  $g[E^y \cup F^y]$  は閉路をもたない. これにより,  $g[E^x]$  および  $g[E^y]$  には, すべての頂点を通るパス (ハミルトンパス) がそれぞれ存在し, 頂点の出現順は一意である.  $g$  の頂点  $v$  に対し, そのハミルトンパス上の出現順を返す関数をそれぞれ  $Ord^x: V \rightarrow N$ ,  $Ord^y: V \rightarrow N$  とする.
- (2)  $g[F^x]$  および  $g[F^y]$  において, 任意の相異なる超辺  $h, h' \in H$  と頂点  $v \in V - V(h)$  に対して  $h$  と  $v$  間および  $h$  と  $h'$  間には必ずパスが存在する.

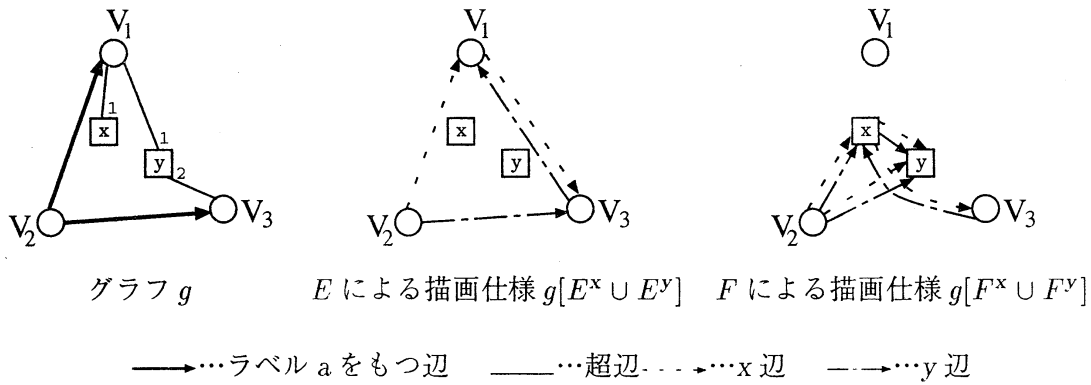


図 1: レイアウト項グラフ  $g$  □は超辺, 超辺の端点はポートを表す.

- (3) 変数ラベル  $x \in X$  をもつ超辺  $h \in H$  において  $|h| = |\text{perm}(h)| = \text{rank}(x)$  である. また  $h$  および  $\text{perm}(h)$  はそれぞれ  $g[E^x], g[E^y]$  上ですべての頂点を通るパスの出現順にそれぞれの要素を並べたリストである.
- (4)  $g[F^s]$  ( $s \in \{x, y\}$ ) において, 超辺  $(u_1, u_2, \dots, u_k)$  から超辺  $(v_1, v_2, \dots, v_l)$  へのパスが存在するならば,  $\text{Ord}^s(u_k) \leq \text{Ord}^s(v_1)$ , または  $k = l$  かつ  $u_i = v_i$  ( $1 \leq i \leq k$ ) である.

辺ラベル  $x, y$  をもつ辺により, 頂点の位置関係を与える.  $g[E^x \cup F^x], g[E^y \cup F^y]$  を描画仕様 (drawing specification) という. 特に必要な場合, グラフ  $g$  の頂点, 超辺, 2 種類の辺の各ラベルづけ関数をそれぞれ  $\varphi_g, \lambda_g, \psi_g, \psi'_g$  で表す. また,  $H = \emptyset$  であるときには, 基礎レイアウト項グラフ (ground layout term graph) という. 以後, 特に断りが無い限り, レイアウト項グラフを単に項グラフ, 基礎レイアウト項グラフを基礎項グラフという. ラベル  $s \in \Lambda$  をもつ辺について, 無向辺は双方向の有向辺の組として扱う. 以後, 図において双方向の有向辺を無向辺で表す. 図 1 にレイアウト項グラフ  $g = (\{v_1, v_2, v_3\}, \{(v_1), (v_1, v_3)\}, \{(v_2, a, v_1), (v_2, a, v_3), (v_2, x, v_1), (v_1, x, v_3), (v_2, y, v_3), (v_3, y, v_1)\}, \{(v_2, x, (v_1)), ((v_1), x, v_3), (v_2, x, (v_1, v_3)), (v_2, y, (v_1)), (v_3, y, (v_1)), (v_2, y, (v_1, v_3)), ((v_1), x, (v_1, v_3)), ((v_1, v_3), y, (v_1))\}, \text{dist}, \{(v_1), (v_3, v_1)\})$  とその描画仕様  $g[E^x \cup E^y], g[F^x \cup F^y]$  を示す.

グラフ  $f$  のすべての頂点を通る  $s$  辺 ( $s \in \{x, y\}$ ) によるパス (これをハミルトン  $s$  パスという) に表れない  $s$  辺  $e = (u, s, v)$  と,  $u$  から  $v$  へのハミルトン  $s$  パスの重みの合計  $W$  に対し,  $W > \text{dist}(e)$  となる辺  $e$  を削除して得られるグラフを  $f'$  とする. このとき項グラフ  $f$  とそれから不要な辺  $e$  を削除したグラフ  $f'$  の関係を  $f \succeq f'$  と表す. また, 明らかに  $|E_f| \geq |E_{f'}|$  である. このとき, 辺の数に関して最小である項グラフ  $f^* = \min\{f_0 \mid f \succeq^* f_0\}$  を  $f$  に関する最小項グラフという.

2 つの項グラフ  $f = (V_f, H_f, E_f, F_f, \text{dist}_f, \text{perm}_f)$  と  $g = (V_g, H_g, E_g, F_g, \text{dist}_g, \text{perm}_g)$  が同型 (isomorphic) であるとは, すべての頂点  $v \in V_f$  において, 以下の条件を満たすような,  $\varphi_f(v) = \varphi_g(\pi(v))$  となる全単射  $\pi: V_f \rightarrow V_g$  が存在する場合であり,  $f \simeq g$  と表す

1.  $f, g$  の最小項グラフ  $f_0, g_0$  とその辺集合  $E_{f_0}, E_{g_0}$  において, 任意の  $(v_1, s, v_2) \in E_{f_0}$  ( $s \in \Lambda \cup \{x, y\}$ ) に対して  $(\pi(v_1), s, \pi(v_2)) \in E_{g_0}$  である.
2. 辺  $(v_1, s, v_2) \in E_{f_0}$  ( $s \in \{x, y\}$ ) に対し,  $\text{dist}_{f_0}((v_1, s, v_2)) = \text{dist}_{g_0}((\pi(v_1), s, \pi(v_2)))$  である.
3. 写像  $\pi^*: V_{f_0}^* \rightarrow V_{g_0}^*$  を  $\pi^*((u_1, \dots, u_r)) = (\pi(u_1), \dots, \pi(u_r))$  と定義するとき, 任意の  $h \in H_{f_0}$  に対し  $\pi^*(h) \in H_{g_0}$  かつ,  $\lambda_{g_0}(\pi^*(h)) = \lambda_{f_0}(h)$  である.
4. 頂点  $v \in V_{f_0}$  と超辺  $h_1, h_2 \in H_{f_0}$  による辺をそれぞれ  $(v, s, h_1), (h_1, s, v), (h_1, s, h_2)$  と表す. このとき, 任意の  $(v, s, h_1) \in F_{f_0}$  に対し  $(\pi(v), s, \pi^*(h_1)) \in F_{g_0}$  かつ任意の  $(h_1, s, v) \in F_{f_0}$  に対し

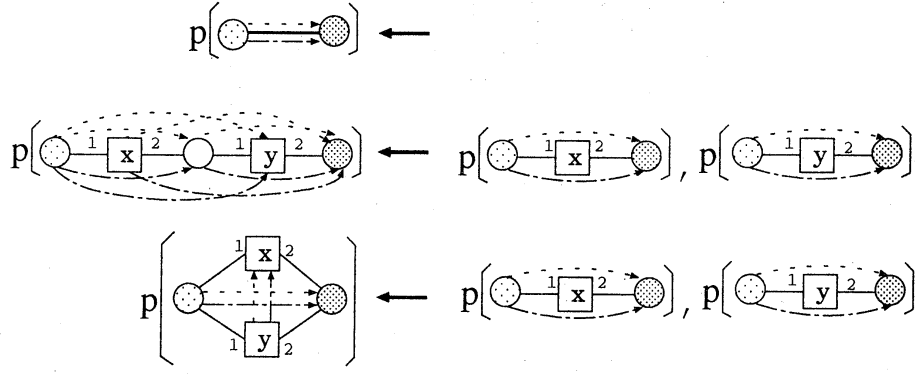


図 2: Layout Formal Graph System

$(\pi^*(h_1), s, \pi(v)) \in F_{g_0}$  かつ任意の  $(h_1, s, h_2) \in F_{f_0}$  に対し  $(\pi^*(h_1), s, \pi^*(h_2)) \in F_{g_0}$  である.

論理プログラミングと同様に、述語記号と項グラフによる形式  $p(g_1, g_2, \dots, g_n)$  をアトム (atom) という。また、 $A, B_1, B_2, \dots, B_n (n \geq 0)$  をアトムとすると、 $A \leftarrow B_1, B_2, \dots, B_n$  をグラフ書き換え規則 (graph rewriting rule) といい、グラフ書き換え規則の有限集合を **Layout Formal Graph System (LFGS)** という。図 2 にその例を示す。

$x$  を変数、 $g$  を項グラフとする。 $g$  の異なる  $rank(x)$  個の頂点をすべて通る  $g[E^x]$  上のパスと  $g[E^y]$  上のパスにおいて、 $Ord^x(u_1) = 1$  かつ  $Ord^y(v_1) = 1$  を満たすように頂点を出現順に並べたリストをそれぞれ  $\sigma^x = (u_1, u_2, \dots, u_k), \sigma^y = (v_1, v_2, \dots, v_k)$  とする。このとき、形式  $x := [g, (\sigma^x, \sigma^y)]$  を束縛 (binding) といい、束縛の有限集合を代入 (substitution) という。項グラフ  $f$  に代入  $\theta = \{x_1 := [g_1, (\sigma_1^x, \sigma_1^y)], \dots, x_n := [g_n, (\sigma_n^x, \sigma_n^y)]\}$  を次のように適用することにより新しい項グラフ (これを  $f\theta$  と表記する) を得る。

1.  $r_i = rank(x_i), N_i = |V_{g_i}|$  とする。 $\theta$  の各束縛  $x_i := [g_i, (\sigma_i^x, \sigma_i^y)] (1 \leq i \leq n)$  に対し、ラベル  $x_i$  をもつ超辺  $t_1, \dots, t_q$  とそれらに付随する  $x, y$  辺を  $f$  から取り除く。
2.  $t_j (1 \leq j \leq q)$  において、 $(\sigma_i^x, \sigma_i^y) = (\pi^*(t_j), \pi^*(perm_f(t_j)))$  となる全単射  $\pi^* : V_f^* \rightarrow V_g^*$  が存在するならば、 $1 \leq m \leq r_i$  に対し  $t_j$  の  $m$  番目の要素と  $\sigma_i^x$  の  $m$  番目の要素を同一視することにより  $f$  に  $g_i$  を埋め込む。
3.  $\sigma^z$  を形成する頂点集合を  $V(\sigma^z)$  とするとき、 $V_{g_i} - \{v \mid v \in V(\sigma^x)\}$  の頂点を  $g_i[E^x]$  上で出現順に並べたリストを  $(v_1^x, \dots, v_{N_i-r_i}^x)$  とする。切り離れた  $t_j$  に付随していた  $F$  の  $x$  辺が  $(t_j, x, u)$  ( $u$  は頂点または超辺) ならば新しく  $x$  辺  $(v_{N_i-r_i}^x, x, u)$  を、切り離れた  $x$  辺が  $(u, x, t_j)$  ならば新しく  $x$  辺  $(u, x, v_1^x)$  を、それぞれ  $E_{f\theta}^x$  に加える。 $y$  辺についても同様に行う。

項グラフ  $f$  に代入  $\theta = \{x := [g, ((u_1, u_2), (u_2, u_1))]\}$  を適用して得られる項グラフ  $f\theta$  の例を図 3 に示す。

グラフの導出は、論理プログラミングと同様に定義する。空ゴールで終わる導出を反駁 (refutation) という。グラフ  $g$  と LFGS  $\Gamma$ 、 $\Gamma$  の述語記号  $p$  が与えられたとき、ゴール  $\leftarrow p(g)$  からの反駁が存在する場合、 $g$  は  $\Gamma$  により生成可能であるという。LFGS  $\Gamma$  とその述語記号  $p$  において、 $\Gamma$  によって生成されたグラフをグラフ言語 (graph language) といい、 $GL(\Gamma, p)$  と表記する。

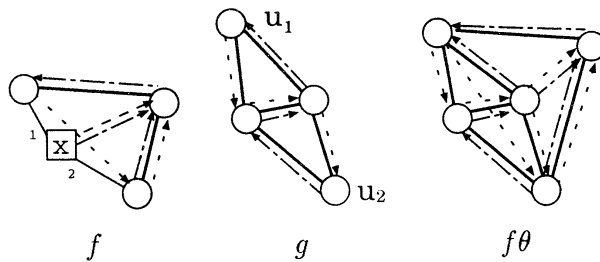


図 3: 項グラフの代入

### 3 レイアウト項グラフの同型問題

LFGS を用いた発見システムを構築する場合に項グラフの同型判定が必要となる．一般的にグラフ同型問題 (2つのグラフ  $f = (V_f, E_f), g = (V_g, E_g)$  において,  $(u, v) \in E_f \Leftrightarrow (\pi(u), \pi(v)) \in E_g$  となるような全単射  $\pi : V_f \rightarrow V_g$  が存在するか否かを決定する問題) は NP には入っているが P に入っているかは未解決な問題の一つである．しかし, レイアウト項グラフにおける同型判定は, 閉路をもたない有向グラフのハミルトンパスを見つける問題に帰着でき, これは多項式時間で解けることが知られている [4]. これより, 以下の定理を得る．

**定理 1** レイアウト項グラフにおける同型問題は, 多項式時間計算可能である．

定理 1 の概説を以下に述べる．項グラフ  $f$  の頂点  $v$  に対し,  $f$  でのハミルトン  $s$  パスでの  $v$  の出現順を返す関数を  $Ord_f^s : V_f \rightarrow N$  とする．すべての  $v \in V_f$  に対して,  $Ord_f^x(v) = Ord_g^x(\pi(v))$  かつ  $Ord_f^y(v) = Ord_g^y(\pi(v))$  を満たすような全単射  $\pi$  が存在するとき,  $\pi$  は  $Ord_f^x, Ord_f^y, Ord_g^x, Ord_g^y$  に無矛盾 (consistent) という．このとき, 2つのレイアウト項グラフ  $f = (V_f, H_f, E_f, F_f, dist_f, perm_f)$  と  $g = (V_g, H_g, E_g, F_g, dist_g, perm_g)$  における同型判定アルゴリズムを以下に示す．

1.  $f, g$  の頂点, 辺, 超辺の要素数が等しいかを調べる
2.  $f, g$  のそれぞれの最小項グラフ  $f_0, g_0$  上でハミルトン  $x$  パス, ハミルトン  $y$  パスを見つけ, 各頂点の出現順位を返す関数  $Ord_{f_0}^x, Ord_{f_0}^y, Ord_{g_0}^x, Ord_{g_0}^y$  を作り, 無矛盾か否かを判定する．
3. 各頂点に対し, 頂点ラベルの一致を調べる．
4. 辺について, 辺の存在, 辺ラベルの一致, 辺の重みの一致を調べる．
5. 超辺のラベルの一致を調べる．

このアルゴリズムにおいて重要な部分は, 2において  $f_0, g_0$  のハミルトン  $x(y)$  パスを見つけることにある．閉路をもたない有向グラフからハミルトンパスを見つける問題は多項式時間で計算可能であることが知られている [4]. 項グラフの定義により, 描画仕様  $f[E^x], f[E^y], g[E^x], g[E^y]$  は閉路をもたない有向グラフなので, ハミルトン  $x(y)$  パスは多項式時間で求めることができる．また, 1, 3-5 については, それぞれ頂点数, 辺数, 超辺数の多項式時間で計算可能である．したがって, レイアウト項グラフにおける同型判定は多項式時間計算可能である．

### 4 レイアウトグラフ文法との比較

レイアウトグラフ文法と LFGS のグラフ生成能力を比較し, LFGS が知識獲得システムでの知識表現に適していることを示す．

レイアウトグラフ文法 (Layout Graph Grammar, LGG) [2] は, 非終端頂点ラベルの集合  $N$ , 終端頂点ラベルの集合  $T$ , 終端辺ラベルの集合  $\Delta$ , 生成規則の集合  $P$ , スタート記号と呼ばれる特別な非終端頂点ラベル  $S$  の組から成るグラフ文法  $GG = (N, T \cup \Delta, P, S)$  に描画仕様を付加したものである．

$N_g(u)$  をグラフ  $g$  において頂点  $u$  に隣接している頂点の集合とする. ラベル  $A$  をもつグラフ  $g$  の頂点  $w$  に生成規則  $p = (A, R, C)$  を適用し, 新しいグラフ  $g'$  を生成するステップを以下に示す.

1. 頂点  $w$  に接続されている辺を  $g$  から取り除く.
2. 頂点  $w$  をグラフ  $R$  で置き換える.
3. 連結関係  $C$  に従い,  $N_g(w)$  の頂点とグラフ  $R$  の頂点間に辺を作成する.

グラフが生成された後, その手順をたどることによって描画仕様を変更し, 頂点のレイアウトを決定する. このような手順を経て, LGG  $G$  によって生成されたグラフ言語を  $L(G)$  と表記する.

LFGS と LGG の生成能力に関し以下の定理が得られる.

**定理 2**  $G$  を LGG とする. このとき,  $GL(\Gamma, p) = L(G)$  となる LFGS  $\Gamma$  とその述語記号  $p$  が存在する.

この定理により LGG で生成できるグラフは LFGS で生成可能である. また, LFGS は LGG では生成することのできない格子や平面的グラフ, 有向非閉路グラフを生成することができる. このことより, LGG に比べ LFGS の方がグラフ生成能力が高いといえる.

帰納論理プログラミング (Inductive Logic Programming, ILP)[3] という学習技法を用いた発見システムが提案されている [5]. LFGS は論理プログラミングを元にしたグラフ生成システムなので, LFGS を知識表現としてもち, ILP を学習手法としてもつ発見システムを構築することができる. これらのことより, 発見システムにおける知識を表現する規則として LFGS が適していると思われる.

## 5 おわりに

本研究では, FGS に描画仕様を付加することにより, 距離空間上のグラフを扱う Layout Formal Graph System を構築し, レイアウト項グラフの同型問題は多項式時間で解けることを示した. また LFGS は LGG よりもグラフ生成能力が高いことを示した. 以上のことより, LFGS は知識発見システムの知識表現に適しているといえる. 今後の課題として, LGG と同等のグラフ生成能力を有する LFGS の部分クラスの同定を行い, LFGS を用いた知識発見システムの開発を行う予定である.

## 参考文献

- [1] S. Arikawa and K. Furukawa (Eds.), Proc.DS-99, Springer-Verlag, LNAI-1721, 1999.
- [2] F.J.Brandenburg, "Designing Graph Drawings by Layout Graph Grammars," Proc. GD-94, Springer-Verlag, LNCS-894, pp.416-427, 1994.
- [3] S.Džeroski, N.Jacobs, M.Molina, C.Moure, D.Muggleton, and W.V.Lear, "Detecting traffic problems with ILP," Proc. ILP-98, Springer-Verlag, LNAI-1446, pp.281-290, 1998.
- [4] M.R. Garey and D.S. Johnson, "Computers and Intractability," W.H.Freeman and company, 1983.
- [5] T.Miyahara, T.Shoudai, T.Uchida, et al., "Discovering New Knowledge from Graph Data Using Inductive Logic Programing" Proc. ILP-99, Springer-Verlag, LNAI-1634, pp.222-233, 1999.
- [6] T. Uchida, T. Shoudai and S. Miyano, "Parallel Algorithms for Refutation Tree Problem on Formal Graph Systems," IEICE Trans. Info. & Syst., Vol. E78-D, No. 2, pp.99-112, 1995.